Amendments to the Claims

Please cancel claim 1 without prejudice or disclaimer.

Please add the following new claims 2-20:

1 ~~2~~. (New) A system for register renaming in a computer system capable of out-of-order instruction execution, comprising:

a temporary buffer comprising a plurality of storage locations for storing execution results, wherein an execution result for an instruction in an instruction window is stored in one of said plurality of storage locations, said one of said plurality of storage locations being assigned to said instruction in said instruction window; and

tag assignment logic that outputs a tag comprising a temporary buffer storage location address in place of a register address for an operand of a first instruction, wherein said temporary buffer storage location address is an address of said operand in one of said plurality of storage locations if said first instruction is dependent on a previous one of said plurality of instructions in said instruction window for said operand.

A1

2 ~~3~~. (New) The register renaming system of claim 1 ~~2~~, further comprising termination logic that transfers said execution results in said plurality of storage locations in said temporary buffer to register file locations in-order based on the order of instructions in said instruction window.

3 ~~4~~. (New) The register renaming system of claim 2 ~~3~~, wherein said termination logic transfers a plurality of execution results from said temporary buffer to said register file simultaneously.

4 ~~5~~. (New) The register renaming system of claim 2 ~~3~~, wherein said termination logic transfers an execution result for an instruction from said temporary buffer to said register

A

file when all execution results for all prior instructions are retirable.

5 ~~6~~. (New) The register renaming system of claim ~~2~~ 1, wherein said tag further comprises an identifier that indicates whether said address within said tag is an address within a register file or within said plurality of storage locations.

6 ~~7~~. (New) The register renaming system of claim ~~2~~ 1, further comprising register file port multiplexers that pass said tags to read address ports of said temporary buffer for accessing said instruction execution results.

7 ~~8~~. (New) A computer system, comprising:

a memory unit for storing program instructions;

a bus coupled to said memory unit for retrieving said program instructions; and

a processor coupled to said bus, wherein said processor comprises a register renaming system, comprising:

> a temporary buffer comprising a plurality of storage locations for storing execution results, wherein an execution result for an instruction in an instruction window is stored in one of said plurality of storage locations, said one of said plurality of storage locations being assigned to said instruction in said instruction window; and
>
> tag assignment logic that outputs a tag comprising a temporary buffer storage location address in place of a register address for an operand of a first instruction if said first instruction is dependent on a previous one of said plurality of instructions in said instruction window for said operand, wherein said temporary buffer storage location address is an address of said operand in one of said plurality of storage locations.

A1

8 ~~9~~. (New) The computer system of claim ~~8~~ 7, wherein said processor further comprises termination logic that transfers said execution results in said plurality of storage locations in said temporary buffer to register file locations in-order based on the order of instructions in said instruction window.

A

~~10~~ 9. (New) The computer system of claim ~~9~~ 8, wherein said termination logic transfers a plurality of execution results from said temporary buffer to said register file simultaneously.

~~11~~ 10. (New) The computer system of claim ~~8~~ 8, wherein said termination logic transfers an execution result for an instruction from said temporary buffer to said register file when all execution results for all prior instructions are retirable.

~~12~~ 11. (New) The computer system of claim ~~8~~ 7, wherein said tag further comprises an identifier that indicates whether said address within said tag is an address within a register file or said plurality of storage locations.

~~13~~ 12. (New) The computer system of claim ~~8~~ 7, wherein said processor further comprises register file port multiplexers that pass said tag to read address ports of said temporary buffer for accessing said execution results.

~~14~~ 13. (New) A method for register renaming, comprising:

storing, in a temporary buffer, out-of-order execution results in storage locations assigned to instructions in an instruction window;

generating at least one tag to specify an address in said temporary buffer at which said out-of-order execution results are temporarily stored; and

outputting one of said at least one tag in place of a register address for an operand of a first instruction if said first instruction is dependent on a previous instruction in said instruction window, wherein said tag comprises an address of said operand in said temporary buffer.

~~15~~ 14. (New) The method of claim ~~14~~ 13, further comprising:

transferring said out-of-order execution results in said temporary buffer to a register file in-order based on the order of instructions in said instruction window.

A

15 ~~16~~. (New) The method of claim 14 ~~15~~, further comprising:

transferring a plurality of execution results from said temporary buffer to said register file simultaneously.

16 ~~17~~. (New) The method of claim 14 ~~15~~, wherein said transferring further comprises transferring an out-of-order execution result from said temporary buffer to said register file when all execution results for all prior instructions are retirable.

A

17 ~~18~~. (New) The method of claim 13 ~~14~~, further comprising:

determining data dependencies between the instructions in said instruction window.

18 ~~19~~. (New) The method of claim 13 ~~14~~, wherein said generating at least one tag comprises generating at least one tag comprising an address and an identifier that indicates whether said address within said tag is an address within a register file or said temporary buffer.

19 ~~20~~. (New) The method of claim 13 ~~14~~, further comprising:

passing said tags to read address ports of said temporary buffer for accessing said out-of-order execution results.

A